

PCT/JP 2004/006517

日 本 国 特 許 庁  
JAPAN PATENT OFFICE

07. 5. 2004

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日            2 0 0 3 年   7 月   9 日  
Date of Application:

出 願 番 号            特 願 2 0 0 3 - 2 7 2 3 3 3  
Application Number:  
[ST. 10/C]:            [ J P 2 0 0 3 - 2 7 2 3 3 3 ]

REC'D 01 JUL 2004

WIPO

PCT

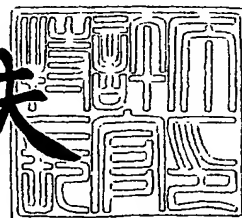
出 願 人            ソニー株式会社  
Applicant(s):

PRIORITY DOCUMENT  
SUBMITTED OR TRANSMITTED IN  
COMPLIANCE WITH  
RULE 17.1(a) OR (b)

2 0 0 4 年   6 月 1 1 日

特許庁長官  
Commissioner,  
Japan Patent Office

今 井 康 夫



出証番号   出証特 2 0 0 4 - 3 0 5 0 4 0 4

【書類名】 特許願  
【整理番号】 0390416503  
【提出日】 平成15年 7月 9日  
【あて先】 特許庁長官殿  
【国際特許分類】 G06F 01/00  
【発明者】  
    【住所又は居所】 東京都品川区北品川 6 丁目 7 番 3 5 号 ソニー株式会社内  
    【氏名】 下山 健  
【発明者】  
    【住所又は居所】 東京都品川区北品川 6 丁目 7 番 3 5 号 ソニー株式会社内  
    【氏名】 目黒 哲正  
【発明者】  
    【住所又は居所】 東京都品川区北品川 6 丁目 7 番 3 5 号 ソニー株式会社内  
    【氏名】 寺西 努  
【発明者】  
    【住所又は居所】 東京都品川区北品川 6 丁目 7 番 3 5 号 ソニー株式会社内  
    【氏名】 中野 靖郎  
【発明者】  
    【住所又は居所】 東京都品川区北品川 6 丁目 7 番 3 5 号 ソニー株式会社内  
    【氏名】 河原 博和  
【特許出願人】  
    【識別番号】 000002185  
    【氏名又は名称】 ソニー株式会社  
【代理人】  
    【識別番号】 100069051  
    【弁理士】  
    【氏名又は名称】 小松 祐治  
    【電話番号】 0335510886  
【選任した代理人】  
    【識別番号】 100116942  
    【弁理士】  
    【氏名又は名称】 岩田 雅信  
    【電話番号】 0335510886  
【先の出願に基づく優先権主張】  
    【出願番号】 特願2003-129219  
    【出願日】 平成15年 5月 7日  
【手数料の表示】  
    【予納台帳番号】 048943  
    【納付金額】 21,000円  
【提出物件の目録】  
    【物件名】 特許請求の範囲 1  
    【物件名】 明細書 1  
    【物件名】 図面 1  
    【物件名】 要約書 1  
    【包括委任状番号】 0117652

**【書類名】 特許請求の範囲****【請求項 1】**

制御対象の稼動状態を観測して、該制御対象の動作周波数を制御する周波数制御装置において、

上記制御対象の稼動状態に応じて上記動作周波数を決定する周波数決定手段と、

上記周波数決定手段により決定される上記動作周波数について、その範囲又は値を制限する周波数制限手段を備えている

ことを特徴とする周波数制御装置。

**【請求項 2】**

請求項 1 に記載の周波数制御装置において、

上記制御対象に供給されてその動作周波数を規定するためのクロック信号の周波数が、予め規定され若しくは指定された周波数範囲又は周波数値でない場合に、上記周波数制限手段によって、上記クロック信号の周波数が上記周波数範囲又は周波数値に制限される

ことを特徴とする周波数制御装置。

**【請求項 3】**

請求項 2 に記載の周波数制御装置において、

上記周波数制限手段が、上記クロック信号の周波数の上限又は下限の閾値を設定する閾値設定手段を備えている

ことを特徴とする周波数制御装置。

**【請求項 4】**

請求項 3 に記載の周波数制御装置において、

上記閾値設定手段として、プログラムの実行により閾値を動的に変更することができる周波数設定用レジスタを有する

ことを特徴とする周波数制御装置。

**【請求項 5】**

請求項 3 に記載の周波数制御装置において、

上記閾値設定手段によって上限及び下限の閾値を同時に設定することにより、上記クロック信号の周波数幅が制限される

ことを特徴とする周波数制御装置。

**【請求項 6】**

請求項 5 に記載の周波数制御装置において、

上記閾値設定手段によって、上限及び下限の閾値を同じ値に設定することにより、上記クロック信号の周波数が一定値に制限される

ことを特徴とする周波数制御装置。

**【請求項 7】**

請求項 2 に記載の周波数制御装置において、

上記周波数制限手段が、上記クロック信号の周波数を指定するための周波数指定手段を備えている

ことを特徴とする周波数制御装置。

**【請求項 8】**

請求項 1 に記載の周波数制御装置において、

一定時間毎に上記制御対象の稼動状態を観測して、該稼動状態に基づいて上記周波数決定手段が動作周波数を決定する

ことを特徴とする周波数制御装置。

**【請求項 9】**

請求項 1 に記載の周波数制御装置において、

上記制御対象が、複数のクロック周波数のうち設定された周波数に基づいて動作する演算処理用デバイスである

ことを特徴とする周波数制御装置。

**【請求項 10】**

随時に可変設定される周波数に基いて動作する対象デバイスの稼動状態を観測して、該対象デバイスに供給されるクロック信号の周波数を制御する情報処理装置において、

上記対象デバイスの稼動状態に応じて上記周波数を決定する周波数決定手段と、

上記周波数決定手段により決定される上記周波数について、その範囲又は値を制限する周波数制限手段を備えている

ことを特徴とする情報処理装置。

【請求項 11】

請求項 10 に記載の情報処理装置において、

上記対象デバイスに供給されるクロック信号の周波数が、予め規定され若しくは指定された周波数範囲又は周波数値でない場合に、上記周波数制限手段によって、上記クロック信号の周波数が上記周波数範囲又は周波数値に制限される

ことを特徴とする情報処理装置。

【請求項 12】

請求項 11 に記載の情報処理装置において、

上記周波数制限手段が、上記クロック信号の周波数の上限又は下限の閾値を設定する閾値設定手段を備えている

ことを特徴とする情報処理装置。

【請求項 13】

請求項 12 に記載の情報処理装置において、

上記閾値設定手段として、プログラムの実行により閾値を動的に変更することができる周波数設定用レジスタを有する

ことを特徴とする情報処理装置。

【請求項 14】

請求項 13 に記載の情報処理装置において、

上記閾値設定手段によって上限及び下限の閾値を同時に設定することにより、上記クロック信号の周波数幅が制限される

ことを特徴とする情報処理装置。

【請求項 15】

請求項 14 に記載の情報処理装置において、

上記閾値設定手段によって、上限及び下限の閾値を同じ値に設定することにより、上記クロック信号の周波数が一定値に制限される

ことを特徴とする情報処理装置。

【請求項 16】

請求項 11 に記載の情報処理装置において、

上記周波数制限手段が、上記クロック信号の周波数を指定するための周波数指定手段を備えている

ことを特徴とする情報処理装置。

【請求項 17】

請求項 10 に記載の情報処理装置において、

一定時間毎に上記対象デバイスの稼動状態を観測して、該稼動状態に基いて上記周波数決定手段がクロック周波数を決定する

ことを特徴とする情報処理装置。

【請求項 18】

請求項 10 に記載の情報処理装置において、

上記対象デバイスが演算処理用デバイスである

ことを特徴とする情報処理装置。

【請求項 19】

制御対象の稼動状態を観測して、該制御対象の動作周波数を制御するための装置に用いられて機能するプログラムにおいて、

負荷の特性又は処理の精度に合わせて上記動作周波数を制限するための制限値を、起動

の際又は起動後に設定するステップを有する  
ことを特徴とするプログラム。

【請求項 20】

請求項 19 に記載したプログラムにおいて、  
起動の際又は起動後に行われた動作周波数の制限に係る設定を解除し又は再設定するステップを有する  
ことを特徴とするプログラム。

【請求項 21】

請求項 4 に記載した周波数制御装置において、  
動作中の各プログラムからその動作に応じた上記クロック信号の周波数について要求される情報の通知を受けて、該情報を統一的に管理する周波数管理手段を設け、該周波数管理手段の設定指令に基づき上記閾値設定手段によって上記クロック信号の周波数範囲又は周波数値が規定される  
ことを特徴とする周波数制御装置。

【請求項 22】

請求項 21 に記載した周波数制御装置において、  
動作中の各プログラムから、上記クロック信号の周波数について要求される下限周波数値又は上限周波数値を含む情報の通知を上記周波数管理手段が受けた場合に、各情報のうち最高の周波数値を採用して上記クロック信号の周波数範囲又は周波数値が決定される  
ことを特徴とする周波数制御装置。

【請求項 23】

請求項 21 に記載した周波数制御装置において、  
動作中の各プログラムから、上記クロック信号の周波数について要求される下限周波数値又は上限周波数値を含む情報の通知を上記周波数管理手段が受けた場合に、各情報の周波数値を加算した周波数値を採用して上記クロック信号の周波数範囲又は周波数値が決定される  
ことを特徴とする周波数制御装置。

【請求項 24】

請求項 21 に記載した周波数制御装置において、  
あるプログラムが終了した場合には、該プログラムの終了が上記周波数管理手段に通知されて、動作中の他のプログラムから上記クロック信号の周波数について要求される下限周波数値又は上限周波数値を含む情報に基づいて上記クロック信号の周波数範囲又は周波数値が決定される  
ことを特徴とする周波数制御装置。

【請求項 25】

請求項 23 に記載した周波数制御装置において、  
上記プログラムに係るプロセス又はスレッドで必要とされる上記クロック信号の周波数が予め登録されており、動作中の各プログラムに関して、新たにプロセス又はスレッドを起動する前に、該プロセス又はスレッドを含む全体に係る周波数の加算総計値を上記周波数管理手段が算出した上で周波数設定を行い、また、起動された上記プロセス又はスレッドが終了した場合にはその終了直後にその周波数を上記周波数管理手段が上記加算総計値から減算する  
ことを特徴とする周波数制御装置。

【請求項 26】

請求項 13 に記載した情報処理装置において、  
動作中の各プログラムからその動作に応じた上記クロック信号の周波数について要求される情報の通知を受けて、該情報を統一的に管理する周波数管理手段を設け、該周波数管理手段の設定指令に基づき上記閾値設定手段によって上記クロック信号の周波数範囲又は周波数値が規定される  
ことを特徴とする情報処理装置。

**【請求項 27】**

請求項 26 に記載した情報処理装置において、

動作中の各プログラムから、上記クロック信号の周波数について要求される下限周波数値又は上限周波数値を含む情報の通知を上記周波数管理手段が受けた場合に、各情報のうち最高の周波数値を採用して上記クロック信号の周波数範囲又は周波数値が決定されることを特徴とする情報処理装置。

**【請求項 28】**

請求項 26 に記載した情報処理装置において、

動作中の各プログラムから、上記クロック信号の周波数について要求される下限周波数値又は上限周波数値を含む情報の通知を上記周波数管理手段が受けた場合に、各情報の周波数値を加算した周波数値を採用して上記クロック信号の周波数範囲又は周波数値が決定される

ことを特徴とする情報処理装置。

**【請求項 29】**

請求項 26 に記載した情報処理装置において、

あるプログラムが終了した場合には、該プログラムの終了が周波数管理手段に通知されて、動作中の他のプログラムから上記クロック信号の周波数について要求される下限周波数値又は上限周波数値を含む情報に基づいて上記クロック信号の周波数範囲又は周波数値が決定される

ことを特徴とする情報処理装置。

**【請求項 30】**

請求項 28 に記載した情報処理装置において、

上記プログラムに係るプロセス又はスレッドで必要とされる上記クロック信号の周波数が予め登録されており、動作中の各プログラムに関して、新たにプロセス又はスレッドを起動する前に、該プロセス又はスレッドを含む全体に係る周波数の加算総計値を上記周波数管理手段が算出した上で周波数設定を行い、また、起動された上記プロセス又はスレッドが終了した場合にはその終了直後にその周波数を上記周波数管理手段が上記加算総計値から減算する

ことを特徴とする情報処理装置。

## 【書類名】明細書

【発明の名称】周波数制御装置、情報処理装置及びプログラム

## 【技術分野】

【0001】

本発明は、制御対象の稼動状態を観測して動作周波数を適正に制御するための技術に関する。

## 【背景技術】

【0002】

電子回路システムにおける消費電力は、該システムを稼動するための電源電圧やクロック周波数に左右される（電源電圧の2乗に比例し、クロックの周波数に比例する。）が、コンピュータ等の情報処理装置では稼動状態や負荷状況が一定ではない。例えば、処理すべきタスクが存在する場合にシステムが動作し、タスク処理後に、別のタスクが存在しない場合に待機状態となるようにシステムステートを遷移させる構成形態が知られている。

【0003】

システムの稼動状態を観測してクロック等を制御することで無駄な消費電力を削減する方法として、システムが待機状態になったことを検出してクロック周波数を低減させる方法（例えば、特許文献1参照）や、バスの負荷状況を計測して所定時間内のバスアクセス率を計算し、該アクセス率に応じてクロックを切り替える方法（例えば、特許文献2参照）が挙げられる。

【0004】

尚、システムが待機状態と動作状態を頻繁に繰り返す場合に、その都度、クロック周波数の切替処理を行う必要が生じる等の問題、あるいは、バスの負荷状況を監視するための回路を必要とすること及び必ずしもバスの負荷状況とシステムの負荷状況とが一致するとは限らず汎用性に問題があること等への対策として、本願出願人は、特願2001-343042号にて、システムの稼動状態の割合に応じてクロック周波数を自動的に変更することができ、システムの消費電力をアクティビティ（所定時間においてターゲット回路が稼動している状態の割合）に応じて最適な制御を実現することで、汎用性が高く、システムの負荷を軽減できるようにした構成形態を提案している。

【0005】

【特許文献1】特開平11-219237号公報

【0006】

【特許文献2】特開平11-184554号公報

## 【発明の開示】

## 【発明が解決しようとする課題】

【0007】

ところで、従来の装置では、システムの稼動状態にとって適正な動作周波数を決定するための制御に関して、設計上の自由度等に問題がある。

【0008】

例えば、ターゲット回路（周波数制御対象）の稼動状態を監視するために観測回路等を設けた構成形態において、クロック周波数の観測結果から一定の回路計算により最適なクロック周波数を算出するためのハードウェア構成を備えている場合に、精度や負荷自体の特性等に対して十分な配慮をもって周波数制御を行うことが難しい。システム（コンピュータ機器等）において処理すべき仕事（タスク）には、とにかく早く終わらせたい仕事（通常の計算処理や検索処理等）や、一定の負荷しかかからない仕事であって、早く終わらせなくてもいい仕事（例えば、音楽再生や動画再生）等、各種の状況が想定されるため、それらの全てについて、最適な周波数を決定するための構成を実現することは殆ど不可能であるか、あるいはハードウェアの実装面において非常に複雑な構成や処理を強いられるため、コストや設計面で問題が残る。つまり、制御対象に係る動作周波数の制御によって高性能化や省電力化の効果を十分に発揮させるためには、時々刻々と変化する状況に応じて適切な動作周波数を決定することが必要であるが、各種状況に応じて周波数決定の考え

方やアルゴリズムを変える必要性が生じたり、大幅な設計変更等を余儀なくされる場合がある（制御ロジックにおいて工夫が必要となる等）。

【0009】

そこで、本発明は、制御対象の稼動状態の観測結果に応じた動作周波数制御において、動作周波数を制限したり許容値等を指定できるように構成し、性能の向上や省電力化等を実現することを課題とする。

【課題を解決するための手段】

【0010】

本発明に係る周波数制御装置は、制御対象の稼動状態（稼動状況）に応じて動作周波数を決定する周波数決定手段と、周波数決定手段により決定される動作周波数についてその範囲又は値を制限する周波数制限手段を備えたものである。

【0011】

また、本発明に係る情報処理装置は、随時に可変設定される周波数に基いて動作する対象デバイスの稼動状態を観測し、該対象デバイスの稼動状態に応じてクロック周波数を決定する周波数決定手段と、周波数決定手段により決定されるクロック周波数について、その範囲又は値を制限する周波数制限手段を備えたものである。

【0012】

さらには、上記構成において下記的手段を設けることができる。

【0013】

- ・クロック信号の周波数の上限又は下限の閾値を設定する閾値設定手段
- ・プログラムの実行により閾値を動的に変更することができる周波数設定用レジスタ
- ・クロック信号の周波数を指定するための周波数指定手段
- ・動作中の各プログラムからその動作に応じたクロック信号の周波数について要求される情報の通知を受けて該情報を統一的に管理する周波数管理手段。

【0014】

従って、これらの発明によれば、制御対象の稼動状態に応じて決定される周波数が許容範囲又は許容値である場合には、該周波数に従って制御を行うが、周波数が許容範囲外である場合等において、該周波数を制限することで、制御の適正化を図り、予期せぬ事態の発生を未然に防止することができる。

【0015】

また、本発明に係るプログラムは、制御対象の稼動状態を観測しその動作周波数を制御するための装置に用いられ、負荷の特性又は処理の精度に合わせて動作周波数を制限するための制限値を、起動の際又は起動後に設定するステップを有している。

【0016】

従って、この発明によれば、プログラムの実行によって周波数を制限することができ、処理すべき仕事の性質等に応じて柔軟な周波数制御を実現することができる。

【発明の効果】

【0017】

本発明に係る周波数制御装置や情報処理装置によれば、制御対象に係る動作周波数について制限を加えるための手段を講じることによって、負荷の特性等に応じて適正な周波数制御が行われるように保証し、性能の向上や省電力化を実現することが可能である。しかも、そのために装置構成の大幅な設計変更等を伴うことがない。

【0018】

また、動作周波数が、予め規定され若しくは指定された周波数範囲を逸脱しないように制限したり、あるいは予期しない周波数値を無効化することができる。

【0019】

そして、上記閾値設定手段を設けることで、クロック周波数の上限や下限を規制することができる。

【0020】

さらには、上記周波数設定用レジスタを設けることにより、ソフトウェアにより閾値設



定を行えるので、制御の柔軟性が高い。また、既存のハードウェア構成を大幅に変更せずに済む。

【0021】

そして、上下限値の同時設定により、クロック周波数の許容幅を随時に制限することができる。

【0022】

また、上下限値を同じ値に設定し、あるいはクロック周波数を特定の値に指定することで、目的別の周波数制御が可能となる（例えば、性能を重視して周波数を高くしたい場合等）。

【0023】

制御対象の稼動状態を定期的に観測することにより、その観測結果に基づいて動作周波数を適正に制御することができる。

【0024】

また、演算処理用デバイスへの適用において、その処理能力を十分に引き出したり、あるいは処理にとって必要十分なクロック周波数を設定することで消費電力を低減させることができる。

【0025】

クロック信号の周波数情報を統一的に管理する周波数管理手段を設けることによって、複数のプログラムを動作させた場合に状況を把握して最適な周波数設定を行うことができる。その際、最も高い周波数値の情報に合わせて周波数を動的に設定し、又は各周波数値を加算した情報に合わせて周波数を動的に設定することができ、プログラムの終了によって周波数範囲又は周波数値の設定を自動的に更新することが可能である。

【0026】

そして、動作中のプログラムに関して、プロセス又はスレッドを起動する前や終了した後に、周波数設定を変更できるように構成することにより、精緻かつ迅速な周波数設定が可能である。

【0027】

本発明に係るプログラムによれば、最適な周波数値や周波数範囲を指定できるので、性能低下や消費電力の増加等を防止することができる。また、プログラムによる動作周波数制限の設定が、該プログラムの終了後の処理に波及して悪影響を及ぼさないようにすることができる。

【発明を実施するための最良の形態】

【0028】

図1は本発明に係る基本構成例を概念的に示したものであり、周波数制御装置1の制御対象2としては、その動作周波数について制御が可能であることが前提とされる。例えば、複数のクロック周波数のうち設定された周波数に基いて動作する演算処理用デバイスとして、CPU（中央処理装置）、コプロセッサあるいは撮像素子等が挙げられる。

【0029】

周波数制御装置1は、制御対象2の稼動状態を観測するために観測手段3を有し、制御対象2の動作周波数を制御する。

【0030】

周波数決定手段4は、観測手段3によって観測される制御対象2の稼動状態に応じて動作周波数を決定するために設けられている。つまり、制御対象2を観測して得られる情報に基いて、周波数決定手段4により、現在の状態にとって最適と推定され又は予測される周波数が算定される（尚、具体的な算定方法については後で詳述する。）。

【0031】

周波数決定手段4の出力は、周波数制限手段5を介して制御手段6に送出される。

【0032】

周波数制限手段5は、周波数決定手段4により決定される動作周波数について、その範囲又は値を制限するために設けられている。即ち、周波数決定手段4によって決定される

動作周波数をそのまま制御手段 6 に通知しても良いか否かについて検閲し、例えば、動作周波数が許容範囲を逸脱している場合には該周波数を制限する。

#### 【0033】

制御対象 2 として CPU 等を想定した場合において、CPU 等へ供給されてその動作周波数を規定するためのクロック信号の周波数（観測結果に基いて周波数決定手段 4 により決定されるクロック周波数）が、予め規定され若しくは指定された周波数範囲又は周波数値でない場合において、周波数制限手段 5 は、クロック信号の周波数を、所定の周波数範囲又は周波数値に制限する役目をもつ。例えば、周波数が許容範囲の上限値を超えていると判断した場合には、周波数を上限値以下に制限し、又は周波数が許容範囲の下限値を下回っていると判断した場合には、周波数を下限値以上に制限する。この場合に、クロック信号の周波数についてその上限又は下限の各閾値、あるいは上限及び下限の両閾値を設定するための閾値設定手段を周波数制限手段 5 が備えている。

#### 【0034】

制御手段 6 は、周波数制限手段 5 の出力を受けて制御対象 2 の動作周波数を制御するものである。例えば、周波数制限手段 5 からの周波数情報をもとにクロック信号（システムクロック）を生成して制御対象 2 に供給するためのクロック信号供給回路等が用いられる。

#### 【0035】

また、電源供給手段 7 は、周波数制限手段 5 の出力を受けて制御対象 2 の動作周波数に見合った電源電圧を、周波数制御装置 1 や該周波数制御装置 1 を含むシステムの各部に供給するものであり、クロック周波数に適合した電源電圧の供給に必要とされる。

#### 【0036】

周波数決定手段 4 はシステムのハードウェアを構成しているが、制御対象 2 に係る動作周波数の決定処理を全てハードウェアに委ね、決定された周波数情報の指示をそのまま制御手段 6 に送ったのでは、周波数制御において柔軟性が得られない。そこで、ハードウェアが決定する周波数情報に対して、例えば、周波数の上限を制限する機構や下限を制限する機構を設け、周波数が許容範囲を逸脱しないように、強制的に周波数を変更してしまいうことが望ましい。そして、これら上限値や下限値については、ソフトウェアにより動的に変更できるように構成し、また、制限機構を有効にするか又は無効にするかをソフトウェアにより指定できるように構成する。そのためには、上限又は下限の閾値設定手段として、プログラムの実行により閾値を動的に変更することができる周波数設定用レジスタを使用すれば良く、各閾値については、プログラムの解釈・実行手段（CPU）からみた場合に、レジスタ値とされ、その値を変更することで上記の機構が実現される。

#### 【0037】

尚、閾値設定の形態には、下記に示す例が挙げられる。

#### 【0038】

- ・ 上限及び下限の閾値を同時に設定して、クロック信号の周波数幅を制限する形態
- ・ 上限及び下限の閾値を同じ値に設定して、クロック信号の周波数を一定値に固定する形態（周波数を固定値にすることで、ハードウェアによる周波数自動変更を、一時的に無効化することができる。）。

#### 【0039】

この他、クロック信号の周波数を指定するための周波数指定手段（周波数指定レジスタ等）を設ける構成を採用した場合には、クロック周波数の値を直接設定し、クロック周波数を決定する周波数決定手段 4 により計算されるクロック周波数を無効化して、該周波数をソフトウェアや外部ハードウェア等で指定することができる。

#### 【0040】

以上の機構により、例えば、以下に示すような制御が可能となる。

#### 【0041】

- ・ 観測手段 3 からの情報に基いて周波数決定手段 4 が現在の周波数を上（下）げると判断した場合でも、それを無効化して、周波数をソフトウェア等で設定可能な一定値より上

(下) げないように制御すること。

【0042】

・観測手段3からの情報に基づいて周波数決定手段4が決定する周波数を、一時的に無効化してソフトウェア等による固定周波数あるいは可変周波数での制御を行うこと。

【0043】

次に、制御対象2の稼動状態を観測して、制御対象2の動作周波数を制御するための装置に用いられて機能するプログラムについて説明する。例えば、ユーティリティやアプリケーション、OS (Operating System) 等の基本ソフトウェアにおいて機能を実現するプログラム等に適用することができる。

【0044】

図2は、アプリケーション(プログラム)の起動時や終了時に、上記制御を行うか否かを判断して、適切な周波数値を設定する場合の処理例を概略的に示している。該アプリケーションを起動することにより、その動作中(プログラム実行中)におけるシステムの動作や性能、消費電力等について最適化することができる。

【0045】

まず、アプリケーションを起動する際には、ステップS1で該アプリケーションの起動要求がなされた後、次ステップS2に示す周波数制限の設定処理に進む。つまり、負荷の特性や処理の精度等に合わせてクロック周波数(動作周波数)を制限するための制限値を設定する。例えば、音楽再生や動画再生等を行う場合において、周波数決定回路によって必要以上にクロック周波数が上がった、下がったりしないように、ソフトウェア処理で一時的に制限することにより、性能の向上や消費電力の低減が可能となる。

【0046】

次ステップS3においてアプリケーションが起動されて、その後動作中となり、所定のプログラム処理が行われる。

【0047】

そして、アプリケーションの終了の際には、ステップS4で該アプリケーションの終了要求がなされた後、次ステップS5に示す周波数制限の再設定又は解除処理に進む。つまり、アプリケーション起動の際に行われた周波数制限を解除したり、あるいは、稼動状態に応じた適正な周波数が得られるように再設定を行う。それからステップS6にてアプリケーションが終了する。

【0048】

尚、本例に示す上記ステップS2、S5の処理については、OSやデバイスドライバ、ライブラリ(サブルーチン群)等によって行われるが、以上の処理については、アプリケーション自身が行うことも可能であり、その場合には、ステップS2とS3の順序が逆になり、アプリケーションの起動後に周波数制限の設定が適時に行われる(また、ステップS5については、アプリケーションの終了時に限らず、処理内容や負荷の軽重等を考慮して必要に応じてアプリケーションの動作中に行うことが可能である。)

【0049】

これらの制御は、例えば、以下に示すようなケースで行われる。

【0050】

・ハードウェアとして一定周波数が必要とされる回路を動作させるときだけに、その周波数を最低周波数に設定して回路を動作させるとともに、回路を動作させないときには、その制限を解除し、さらに低い周波数の設定を行えるようにする場合。

【0051】

・低い(又は高い)周波数領域では動作しないハードウェアに対して、動作周波数を適切な設定範囲に制限する場合。

【0052】

・オーディオ信号やビデオ信号のデコード又はエンコード等のように、ソフトウェアとして最低動作周波数の要求や、最高動作周波数の要求が明らかであるものを動作させるときに、周波数制限を行うことにより、周波数の不用意な変更を防ぐ場合や、アプリケーション

ン動作時に周波数が不用意に上昇又は下降しないように防止する場合。

【0053】

・性能や処理速度等が要求されるようなアプリケーションを動作させる場合に、所望の周波数を指定したり、最低周波数（下限）を高い値に設定することによって、消費電力の増加を容認してまでも性能を向上させたい場合（例えば、装置への着脱が可能な半導体メモリ媒体へのデータ書き込みやデータ読み出し等）。

【実施例】

【0054】

本発明に係る実施の一例について説明する。尚、本例に示す情報処理装置としては、例えば、コンピュータ機器や、PDA(Personal Digital Assistance)、移動体通信端末装置、映像機器や撮像機器等への適用が可能である。

【0055】

図3は、情報処理装置10の要部について構成例を示したブロック図である。

【0056】

制御対象であるターゲット回路には、CPU11が用いられており、上記観測手段3を構成するタイマ12、カウンタ13と、上記周波数決定手段4を構成する制御回路14が設けられている。また、クロック供給回路15が上記制御手段6を構成し、電源電圧供給回路16が上記電源供給手段7を構成している。尚、本例ではタイマ12、カウンタ13、制御回路14、クロック供給回路15により周波数制御回路が構成される。

【0057】

CPU11は、複数のクロック周波数のうち随時設定される周波数に基いて動作し、後述するように、多階調（段階）のクロック周波数を生成することができるクロック供給回路15から供給されるシステムクロック「SYSCLK」に同期して動作する（例えば、待機状態と稼働状態を繰り返しながら、所望のタスクを処理する。）。

【0058】

CPU11は、これが動作状態にあるときに、アクティブであることを示す1ビットのフラグ（以下、「アクティビティフラグ」と呼び、「ACFLG」と記す。）がセットされるアクティビティフラグレジスタ111を有する。アクティビティフラグレジスタ111には、例えば、CPU11が稼働している状態のときに論理値「1」がセットされ、待機状態のときに非アクティブを示す論理値「0」がセットされる。

【0059】

そして、CPU11は、ACFLGレジスタ111にセットされるACFLGをカウンタ13に送ることにより、その稼働状態を通知する。

【0060】

CPU11のアクティビティ（所定時間における稼働状態の割合）を計測するための計測時間Tは、タイマ12により計測される。

【0061】

タイマ12は、周波数が固定のクロック信号（これを「FXCLK」と記す。）に同期して動作することにより、一定の時間を計測し、計測結果をその都度、信号「S12」としてカウンタ13及び制御回路14に出力する。

【0062】

図4はタイマ12について構成例を示す図であり、カウンタ121、コンペアレジスタ122、コンパレータ123を有している。

【0063】

カウンタ121は、「CLK」で示すクロック信号を計数して、そのカウント値（これを「VCNT」と記す。）をコンパレータ123に送る。また、コンペアレジスタ122で保持される比較値（これを「VCMP」と記す。）がコンパレータ123に送られ、カウント値VCNTと比較値VCMPとが比較される。

【0064】

タイマ12において、比較値VCMPは制御回路14からの信号S141により設定さ

れ、カウント値VCNTが比較値VCMPと一致した時点で、該一致を報知する一致信号S12がコンパレータ123からカウンタ13及び制御回路14に出力されるとともに、該信号S12がカウンタ121のリセット端子に送られてカウント値がクリアされる。こうして、比較値VCMPに相当する計測時間Tに従い、一定の周期で一致信号S12が出力される。

【0065】

図3に示すカウンタ13は、CPU11のアクティビティを計測するための回路であり、システムクロックSYCLKに同期して動作し、CPU11のACFLGに従い、CPU11が動作しているときにSYCLKのパルスのカウントし、このカウント値を信号「S13」として制御回路14に出力する。

【0066】

図5はカウンタ13の構成例を示す図である。

【0067】

ACFLGによるイネーブル(EN)信号がカウンタ13に与えられ、ACFLGが論理値「1」を示す状態のときに、SYCLKに同期してアップカウント動作が行われる(加算)。また、カウンタ13は、タイマ12からの一致信号S12をクリア信号として受けて、一致信号S12が入力される度にカウント値が初期化される。

【0068】

図6はカウンタ13の動作を説明するためのタイミングチャート図であり、SYCLKを(A)図に、ACFLGを(B)図に、カウンタ13の状態を(C)図にそれぞれ示している。

【0069】

カウンタ13では、図6(C)に示すように、タイマ12から計測時間Tごとに送られてくる一致信号S12に従い、カウント値が初期化される。

【0070】

図6(B)に示すように、CPU11によるACFLGの値が、CPU11の動作状態を示しているとき(本例では、ACFLGが論理値「1」を示すとき)にカウント動作(加算処理)が行われる。

【0071】

一方、CPU11によるACFLGの値が、CPU11の待機状態を示しているとき(本例では、ACFLGが論理値「0」を示すとき)にはカウント動作が停止される。

【0072】

図3に示す制御回路14は、一定時間毎に観測されるCPU11の稼動状態に基づいてSYCLKの周波数を決定するために設けられている。本回路は、タイマ12からの一致信号S12に同期してカウンタ13からの信号S13をとりこみ、計測時間TにおけるCPU11のアクティビティ(これを「ACT」と記す。)を計測する。そして、計測したACTに基づき次に設定すべきSYCLKの周波数を決定する。

【0073】

CPU11の稼動状態に応じて決定されるクロック周波数に対して、その上限や下限の閾値を設定するための閾値設定手段17が設けられており、上記周波数制限手段5を構成している。本例では、プログラムの実行により閾値を動的に変更することができる周波数設定用レジスタ17U、17Lが設けられている。つまり、レジスタ17Uが上限周波数設定用とされ、レジスタ17Lが下限周波数設定用とされ、CPU11によるコマンド実行結果として出力ポート経由で設定される、それらのレジスタ値が制御回路14で参照される。周波数の上限や下限を指定するレジスタを設けることにより、周波数制御回路において算出される周波数の上限や下限を制限する(クロック周波数を、上限及び下限に示す範囲内に規制する)。あるいは、上限と下限を同じ値に規定して特定の周波数が得られるように一時的に固定するといった制御を、ソフトウェア処理によって行えるので、状況に応じた柔軟な周波数制御が可能となる。

【0074】

制御回路14は、こうして決定された周波数をもつSYSCLKをCPU11やカウンタ13に供給するための制御信号「S142」をクロック供給回路15に出力し、また、上述したように、タイマ12のコンペアレジスタ122に設定すべき比較値VCMPの信号S141をタイマ12に出力する。

#### 【0075】

尚、クロック周波数をソフトウェアにより直接指定するための周波数指定手段18として、周波数指定用レジスタを設けた構成形態では、CPU11によるコマンド実行結果として設定されるレジスタ値に対応する周波数が制御回路14において決定され、この場合、ACTの計測結果に基づいて決定される周波数については無効化される（図3には便宜上、レジスタ17U、17Lを用いる形態と、レジスタ18を用いる形態について併せて示している。）。但し、周波数を上げたい場合に指定された周波数値にいきなり変更できないときには、指定周波数値を徐々に変化させて目的値に近づける等の配慮が必要である（このことは、レジスタ17U、17Lを用いて特定の周波数を指定する形態でも同様であり、周波数幅を徐々に狭めていってゼロに漸近させるといった処理を行えば良い）。

#### 【0076】

また、上記のような周波数制限や周波数指定に係る設定をソフトウェアにより解除するためには、例えば、レジスタ値として特別な値をセットするか、あるいは数値的に制限の意味をもたない値（最高許容周波数を超える値やゼロ等）をセットする等の方法を用いれば良い。

#### 【0077】

クロック供給回路15は、異なる周波数のクロック信号を生成してCPU11等へ送出する回路である。本回路はSYSCLKとして多階調（段階）のクロック周波数の信号を生成することが可能であり、制御回路14からの制御信号S142により指示されたクロック周波数をもつSYSCLKを、CPU11、カウンタ13に供給する。尚、多階調のクロック周波数を得るための構成としては、周波数に係る複数の選択肢（クロック信号）から1つを選択する形態や、クロック発振回路における周波数値の設定を任意に変更できるようにした形態、例えば、複数の周波数を時間的に組み合わせて各周波数のクロック信号を決められた時間間隔で切り替える（所定時間に亘る平均周波数値を制御する）形態等が挙げられる。

#### 【0078】

図7はクロック供給回路15の構成例を示す図である。

#### 【0079】

本例では、所定周波数のクロック信号を発振する位相同期ループ回路（PLL回路）151と、分周器152、セクタ153が用いられる。

#### 【0080】

PLL回路151から分周器152に送られたクロック信号は、異なる分周比に従って、複数（本例では、4つのクロック周波数 $f_0$ 、 $f_1$ 、 $f_2$ 、 $f_3$ を示す。）のクロック信号が生成される。

#### 【0081】

セクタ153は、制御回路14からの制御信号S142により指示されたクロック周波数をもつSYSCLKを選択して出力する。

#### 【0082】

例えば、PLL回路151から分周器152に、周波数300MHzのクロック信号が供給され、分周比が、 $1/3$ 、 $1/4$ 、 $1/6$ 、 $1/12$ にそれぞれ規定されている場合、クロック分周により、 $f_0$ （＝100MHz）、 $f_1$ （＝75MHz）、 $f_2$ （＝50MHz）、 $f_3$ （＝25MHz）のクロック周波数のいずれかがセクタ153で選択され、SYSCLKとして出力される。尚、各周波数を組み合わせた制御では、例えば、 $f_1$ を $t_1$ 秒間選択し、 $f_2$ を $t_2$ 秒間選択することを繰り返し行うことで、「 $(f_1 \cdot t_1 + f_2 \cdot t_2) / (t_1 + t_2)$ 」といった中間的な周波数を設定することが可能である。

## 【0083】

制御回路14の制御信号S142は、クロック供給回路15の他、電源電圧供給回路16(図3参照)にも送られる。つまり、クロック供給回路15に指示された周波数においてシステムの動作を保証するために必要な最低の電源電圧(これを「VDD」と記す。)をシステムの各部に供給する(制御信号S142の指示に従い、電源電圧VDDを変化させ、あるいは維持する。)

## 【0084】

次に、制御回路14において、システムの周波数決定にある程度のマージンを考慮し、クロック供給回路15で多階調の周波数の刻みを考慮した場合の動作について、先ず、図8のフローチャート図に従って説明する。

## 【0085】

CPU11は、多階調のクロック周波数を生成することができるクロック供給回路15から供給されるシステムクロックSYSCLKに同期して動作し、待機状態と稼動状態を繰り返しながら、所望のタスクを処理している。先ず、ステップST1では、上記アクティビティACTを計測する。

## 【0086】

処理手順をまとめると下記のようになる。

## 【0087】

(1) CPU11が動作状態にあるときに、ACFLGがレジスタ111にセット(論理値「1」)され、また、待機状態のときには、レジスタ111にACFLGがクリア(論理値「0」)される

(2) レジスタ111の値がカウンタ13に送出されて稼動状態が通知される

(3) タイマ12からの計測時間Tを示す信号S12を受けて、カウンタ13では、ACFLGの示す値に従って計数動作を行う(即ち、CPU11が動作状態の期間中、SYSCLKをカウントする。)

(4) カウント値を制御回路14に出力し、制御回路14が、所定の計測時間TにおいてCPU11が動作していた時間の占める割合(つまり、システムのACT)を算出する。

## 【0088】

次のステップST2では、得られたアクティビティACTを第一の閾値(現在の周波数「fc」からマージン「MGN1」を差し引いた値)と比較し、ACTが該閾値よりも大きいと判別した場合に、ステップST3に進むが、そうでない場合にはステップST4に進む。尚、ACT、fcの比較時には、ACTがfcの関数であることを考慮して周波数に換算するか(ACTを百分率で表す場合に、 $ACT = 100\%$ が、計測期間Tにおいてfcのクロックを全てカウントした数に相当する。)、あるいはACTが計測時間Tでのクロックカウント値を表す場合において、fcを計測時間Tでのクロックカウント値に換算する等、両者を同等の立場で比較する必要があることは勿論である。

## 【0089】

ステップST3では、SYSCLKの周波数を上げる必要があるため、現在の周波数「fc」に対して所定の周波数刻み分(これを「 $\Delta f$ 」と記す。)を加算した周波数値を採用する。つまり、現在の周波数fcよりも1段階上げた周波数( $fc + \Delta f$ )を次の周波数とする決定が行われた後で、ステップST7に進む。

## 【0090】

また、ステップST4では、前ステップST1で得られたACTを第二の閾値(現在の周波数「fc」から周波数刻み分 $\Delta f$ を引算して1段階下げた周波数からさらにマージン「MGN2」を差し引いた値)と比較し、ACTが該閾値よりも小さいと判別した場合にステップST5に進み、そうでない場合にはステップST6に進む。

## 【0091】

ステップST5では、SYSCLKの周波数を下げる必要があるため、現在の周波数fcに対して所定の周波数刻み分 $\Delta f$ を引算した周波数値を採用する。つまり、現在の周波



数  $f_c$  よりも 1 段階下げた周波数 ( $f_c - \Delta f$ ) を次の周波数とする決定が行われた後で、ステップ S T 7 に進む。

【0092】

ステップ S T 6 に到達するのは、ステップ S T 2 および S T 4 のいずれの条件も満たさない場合であり、A C T が S Y S C L K の周波数に対して適切な状態にあることを意味する。よって、現在の周波数 ( $f_c$ ) を維持したままで、ステップ S T 7 に進む。

【0093】

上記ステップ S T 3、S T 5、S T 6 の後、ステップ S T 7 では、変更又は維持された周波数値を、上記レジスタ (図 3 の 17、18 参照) によって規定される周波数範囲や周波数値と比較することでチェックする。そして、周波数値が許容される場合にはステップ S T 1 に戻るが、許容されない場合にはステップ S T 8 に進む。

【0094】

ステップ S T 8 では周波数値について上限又は下限を制限し、あるいは特定の周波数値が得られるように制限した後、ステップ S T 1 に戻る。

【0095】

こうして決定される周波数値に応じて、クロック供給回路 15 では、制御回路 14 からの制御信号 S 142 によりクロック信号が指示される。即ち、該当するクロック周波数をもつ S Y S C L K が選択されて C P U 11、カウンタ 13 に供給される。

【0096】

このように、所定の計測時間 T を周期として周波数の変更又は維持が繰り返されることになる。

【0097】

尚、システムクロックに関して、現在の周波数をもとに次の周波数を決定するためには、周波数を上げるための条件及び上限と次の周波数の与え方、あるいは周波数を下げるための条件及び下限と次の周波数の与え方を、システムの構成や特性等に応じて適切に規定してやれば良い。また、マージン M G N 1、M G N 2 の設定については、両者を同じ値にする方法や両者を異なる値にする方法 (例えば、マージンを A C T や周波数範囲等に応じて段階的又は連続的に変化させる等。) を適宜に用いることが可能である。

【0098】

図 9 は制御回路 14 の構成について一例を示したものである。尚、本例では上記「M G N 1」と「M G N 2」を同じ値「M G N」(正值)としている。

【0099】

本例では、制御回路 14 が、加算器 141、減算器 142、加算器 143、比較器 144、145、セクタ 146、リミッタ 147、記憶素子 148、デコーダ 149 から構成されている。

【0100】

記憶素子 148 には現在の周波数  $f_c$  を示すデータが保持されており、該データが加算器 141、減算器 142、セクタ 146、比較器 145 にそれぞれ送出される。

【0101】

加算器 141 は、 $f_c$  と  $\Delta f$  との加算値「 $f_c + \Delta f$ 」を求めて、これをセクタ 146 に対して出力し、また、減算器 142 は、 $f_c$  から  $\Delta f$  を引算して「 $f_c - \Delta f$ 」を求めて、これをセクタ 146、比較器 144 に対して出力する。

【0102】

加算器 143 は、前記カウンタ 13 の出力する A C T と M G N を加算して、「A C T + M G N」を比較器 144、145 に送出する。

【0103】

比較器 144 は、減算器 142 が出力する「 $f_c - \Delta f$ 」と、加算器 143 が出力する「A C T + M G N」について大小関係を比較し、比較結果をセクタ 146 に送出する。

【0104】

また、比較器 145 は、現在の周波数  $f_c$  と、加算器 143 が出力する「A C T + M G



N]について大小関係を比較し、比較結果をセクタ146に送出する。

【0105】

セクタ146には、 $f_c$ 、「 $f_c + \Delta f$ 」、「 $f_c - \Delta f$ 」の各データが供給され、比較器144、145によるそれぞれの比較結果に基づいて周波数を選択して、リミッタ147に送出する。

【0106】

リミッタ147は、前記したレジスタ(17又は18)による周波数値(上下限值や特定周波数値)を参照して、セクタ146からの周波数値を制限し、その結果を記憶素子148に送る。記憶素子148に新たな周波数のデータが格納される(これにより $f_c$ の値が更新されるが、これは前記タイマ12からの指示に従って計測時間T毎に行われる)。

【0107】

記憶素子148に格納されている $f_c$ のデータはデコーダ149に送られ、ここでは該データをもとに、前記クロック供給回路15においてクロックを選択するための指示信号(S142)を生成して出力する。

【0108】

従って、図8のST2の比較判断が比較器145で行われ、ST4の比較判断が比較器144で行われ、また、ステップST3、ST5、ST6については、セクタ146により行われる。そして、ステップST7、ST8については、レジスタ値を基にしてリミッタ147で処理される。

【0109】

以上説明したように、CPU11はSYSCLOCKに同期して、待機状態や動作状態を繰り返しながらタスクを処理し、ACFLGをカウンタ13に送ることにより、計測時間TでのCPUの稼働状態を表すアクティビティが計測される。そして、計測されたアクティビティに基づき次に設定すべきSYSCLOCKの周波数が決定されることになるが、その際には、クロック周波数がレジスタで指定される上限閾値や下限閾値により制限されたり、あるいは特定の周波数値となるように制限される。こうしてチェックされた周波数をもつSYSCLOCKがクロック供給回路15により発生されてCPU11等へ送られる。従って、周波数制限を受けない場合には、システムのアクティビティ(稼働状態の割合)に応じてクロック周波数が自動的に変更され、また周波数制限を受ける場合には、許容範囲外の周波数が上限値又は下限値に規制され、あるいは、無効化されて特定の周波数に変更される。

【0110】

尚、上記の説明では、制御対象としてCPU11だけを取り上げたが、クロック周波数を制御する対象が複数の場合でも構わない。例えば、CPU11と、複数の周辺回路の稼働状態をそれぞれ観測した結果に基づいてSYSCLOCKの周波数制御を行うことができる。実際の回路システムでは、CPU11と他の周辺回路が同時に動作している状態もあれば、CPU11は動作しているが周辺回路が待機状態であったり、逆に、CPU11が待機状態であって周辺回路が動作している場合もある。

【0111】

そこで、CPUや周辺回路のそれぞれの稼働状態を示す信号を用いて、その論理和(OR)信号を求め、該信号をカウンタ13にイネーブル信号として供給すれば、複数の制御対象について稼働状態を総合的に計測した結果に基づいてSYSCLOCKの周波数制御を行える。つまり、制御対象毎の稼働状態をそれぞれに示すACFLGを用いて、各フラグの論理和信号をカウンタ13に送ることで、該論理和信号が示すフラグについてシステム全体の稼働状態を示すACFLGとみなすことができる。これによって、制御対象の数や種類によらず、同一の制御方法を採用することができるので、汎用性が高い。また、各制御対象の稼働状態を1ビットのACFLGの値で管理することができるので、大幅な回路変更等を伴うことがない。

【0112】

次に、複数のアプリケーションが動作している場合におけるクロック周波数の管理について説明する。従来の周波数管理においては、例えば、アプリケーションに対応した単純な周波数の指定等が可能であるにすぎず、マルチタスク機能をもつOSを搭載した装置等で複数のアプリケーションを動作させた場合に、十分な性能を得ることが困難とされ、また、アプリケーションの起動や終了のタイミングをトリガにして周波数の設定等が行われる程度であって大まかな管理であったために、詳細な周波数の管理を行うことが困難である。

#### 【0113】

そこで、以下では、動作中のプログラムからその動作や処理に応じたクロック周波数に係る要求情報の通知を受けて、該情報を統一的に管理する手段（周波数管理手段）を設けた構成形態について説明する。尚、周波数管理手段（以下、「周波数マネージャ」という。）は、例えば、ソフトウェアとして構成され（OSの一部やデバイスドライバ等）、制御できる設定上のパラメータとして、システムクロックの周波数や、ハードウェアで自動制御されるクロック周波数の上限周波数、下限周波数等が挙げられる。

#### 【0114】

図10は、制御アルゴリズムの一例について説明するための図である。

#### 【0115】

本例では2つのアプリケーションプログラム「PA」と「PB」が時間的にオーバーラップする期間（以下、「重複期間」という。）をもって動作している様子を示しており、各プログラムから呼び出されるライブラリ（又はデバイスドライバ等）8を介して周波数マネージャ9にシステムクロックの周波数要求が通知される。また、図中に示す「TA1」の時点においてPAからライブラリ等がコールされ、「TA2」の時点で呼び出した処理が終了し、また、「TB1」の時点においてPBからライブラリ等がコールされ、「TB2」の時点で呼び出した処理が終了する。尚、TA1やTB1については、各プログラムが起動された時点でも良いし、また、特定の処理が必要となった時点でも良い。

#### 【0116】

本システム上で動くアプリケーションPA、PBから、動画処理や音声処理等の特定の処理においてライブラリやデバイスドライバが呼ばれた場合に、各ライブラリやデバイスドライバは、処理に必要な周波数情報を周波数マネージャ9に与え、そこで最適な設定を行ってもらう。尚、ライブラリやデバイスドライバは、自分の動作内容を事前に把握しているので、その動作に応じた最適な設定パラメータの情報を数値あるいは参照テーブル等の形態で保有することができる。

#### 【0117】

システムクロック情報等の設定については、以下の例が挙げられる。

#### 【0118】

- ・最低限必要な周波数があるアプリケーションにおいて、最低の周波数を設定すること（例えば、シリアル通信を行う場合に周波数を16MHz（メガヘルツ）以上にする等。）
- ・性能を上げるために一時的に周波数の下限値を高くすること（例えば、半導体メモリ等を用いた2次記憶装置を使用する場合に、周波数を96MHz以上にする等。）
- ・性能やデータ品質の低下等を防止するために周波数が下限値未満にならないようにすること（例えば、PCM音声データのミキシング時に周波数を32MHz以上にしたり、動画の録画時に周波数を64MHz以上にする等。）
- ・一定以上の性能を必要としないアプリケーションにおいて、最高の周波数を設定すること（例えば、音声データの再生時に周波数を64MHz以下にする等。）
- ・アプリケーションに見合った固定の周波数値あるいは周波数の初期値を設定すること。

#### 【0119】

この他、処理対象であるデータの種類や転送レート、フォーマット等に応じて周波数の下限値又は上限値を設定することができる。動画像データや音声データを含むファイルを

開く場合等においてそのビットレートに応じてCPUやDSP (Digital Signal Processor) 等の動作周波数の下限値を規定する場合に、例えば、MPEG (Moving Picture Experts Group) 4のビットレート (786 kbps) に合わせて周波数下限値 (123 MHz) を規定することが好ましい (ビットレートが高いほど下限値を高くする。)

#### 【0120】

システム全体で見た場合、複数のアプリケーションが同時に処理される (フォアグラウンドプロセスに対して、バックグラウンドプロセスが背後にまわりながら動く。) 状況下で、アプリケーションやライブラリ又はデバイスドライバ等からは、現時点で他にどのようなプログラムが実行されているかが分からないため、個別的な周波数設定を直接行ったのでは、最適な設定を行うことができない。従って、統一的な周波数の管理を行う手段が必要であり、周波数マネージャ9は、周波数の設定情報に基づいて閾値設定を変更することによって周波数範囲や周波数値を規定する役目を有する。

#### 【0121】

図10に示す例では、最低周波数64 MHzのプログラムPAと最低周波数32 MHzのプログラムPBが、TB1の時点からTA2の時点にかけてオーバーラップして動いている状況を表している。各プログラムから呼び出されたライブラリ (又はデバイスドライバ) 8は、その入口と出口において最適な周波数設定情報を周波数マネージャ9に伝える。そして、周波数マネージャ9はこれらの情報をもとに最適な設定を行う。

#### 【0122】

例えば、TA1の時点からTB1の時点にかけての期間において最低周波数が64 MHzに設定され、また、TA2の時点からTB2にかけての期間において最低周波数32 MHzに設定される。尚、図中の「通常動作」とは、周波数マネージャ9やOS等のように、システムにとって最低限必要な要素が動作している状態 (アプリケーション等が動作していない状態での最低周波数の設定状態) を意味する。

#### 【0123】

複数のプログラムが動作している重複期間において、周波数マネージャ9が適正な周波数を設定するためのアルゴリズムとして、例えば、以下に示す2つの形態が挙げられる。

#### 【0124】

(I) 指示された最低周波数 (下限値) 又は最高周波数 (上限値)、あるいは特定の周波数の中でもっとも大きい情報を採用して設定する形態 (最大方式)

(II) 指示された最低周波数 (下限値) 又は最高周波数 (上限値)、あるいは特定の周波数について、それぞれを加算した情報を採用して設定する形態 (加算あるいは総和方式)。

#### 【0125】

先ず、(I) では、クロック周波数について要求される下限周波数値又は上限周波数値を含む情報の通知を周波数マネージャ9が受けた場合に、各情報のうち最高の周波数値を採用する。例えば、動作中の各アプリケーションからライブラリ等を介して要求される下限又は上限の周波数を「 $f_i$ 」 ( $i=1, 2, \dots$ ) と記し、最大値関数を「 $\text{MAX}()$ 」と記すとき、「 $f_s = \text{MAX}(f_1, f_2, \dots)$ 」によって設定値「 $f_s$ 」を算定する。

#### 【0126】

図10の例に (I) を適用した場合、TB1の時点からTA2の時点にかけての重複期間において、64 MHzと32 MHzの領域が重なっており、「 $\text{MAX}(64, 32) = 64$  (MHz)」が設定される。

#### 【0127】

また、(II) では、クロック周波数について要求される下限値又は上限値を含む情報の通知を周波数マネージャ9が受けた場合に、各情報の周波数値を加算した周波数値を採用する。例えば、上記「 $f_i$ 」 ( $i=1, 2, \dots$ ) を用いて、「 $f_s = \sum f_i$ 」 (「 $\sum$ 」は自然数変数「 $i$ 」についての総和を示す。) によって設定値「 $f_s$ 」を算定する。あるいは、さらに一般化して各「 $f_i$ 」に加重係数「 $k_i$ 」を乗じた値を加算する (「 $f_s = \sum (k_i \cdot f_i)$ 」) といった形態でも構わない。

## 【0128】

図10の例に(II)を適用した場合、TB1の時点からTA2の時点にかけての重複期間において、「 $64 + 32 = 96$  (MHz)」が設定される。

## 【0129】

尚、(I)と(II)を状況や条件等に応じて併用した形態も勿論可能であり、各種の算定方法に従ってクロック信号の周波数範囲(上下限值)や周波数値が決定される。また、システムにおいて、あるプログラムが終了した場合には、該プログラムの終了が周波数マネージャ9に通知され、必要に応じて周波数設定が更新されることで、最適な設定に戻す処理が行われる。例えば、(I)の場合には、あるプログラムの終了によって、残るプログラムの周波数情報のうち、最大値が採用され、また、(II)の場合には、あるプログラムの終了によって該プログラムの周波数情報を、それまでの総和から減算したものが採用される。いずれにしても、動作中の他のプログラムからクロック周波数について要求される下限周波数値又は上限周波数値を含む情報に基づいて周波数範囲又は周波数値が決定されることになる。尚、(II)については、複数のプログラムからのリソース(資源)獲得の要求が競合する場合に用いることが好ましく、また、そのような競合の心配がない場合において(I)を用いることが可能である。

## 【0130】

これまでの周波数管理では、複数のアプリケーションが動作することを考慮しておらず、例えば、ある特定のアプリケーションが実行された場合に、該アプリケーションに固有の設定が行われる程度とされ、よって、上記したような緻密な設定を行うことができなかったが、ライブラリやデバイスドライバ等で高い処理性能が必要となる部分において周波数設定の要求を周波数マネージャ9に出せるので、アプリケーションにおける処理の一部(タスク)で性能が要求される状況等にも対応できる。また、複数の性能を要求するアプリケーションが動作していても、周波数マネージャ9によって統一的に周波数の管理を行えるので、アプリケーション側からみた個別的な観点では困難とされる、システム全体に亘る最適設定を実現することが可能である(全体の状況を十分に把握した上で周波数を設定することができる。)

## 【0131】

以上に説明したように、システムクロック等を制御する場合に、例えば、クロック周波数の上下限をソフトウェアで制御できるシステムにおいて、アプリケーションを実行する際に該アプリケーションから呼ばれるライブラリやデバイスドライバ等が、その動作に合わせたシステムクロック情報を周波数マネージャ9に送信して、該周波数マネージャ9がシステム全体の周波数を制御することが好ましい。

## 【0132】

尚、上記の説明では、アプリケーションからライブラリやデバイスドライバを介して周波数マネージャ9と通信する形態を示したが、これに限らず、アプリケーション自身が周波数マネージャ9と通信するといった各種形態での実施が可能である。

## 【0133】

また、アプリケーション、あるいはアプリケーションから呼び出されるライブラリ等を処理単位として周波数の要求情報を周波数マネージャ9に送信する形態には限られず、例えば、プロセス(コンピュータ資源の割り当て単位であり、実行プログラムやスレッド等を含む。)やスレッド(プロセッサの割り当ての最小単位であり、プログラムカウンタや各種レジスタセット、スタック領域、プライベートな記憶領域等で構成される。)を単位として、さらに緻密な制御が可能である。

## 【0134】

消費電力を低減する場合に、CPUやDSP、メモリ、バス等のクロック周波数を必要に応じて変化させる技術が使われるが、マルチスレッド環境(1つのプロセス内で複数のスレッドをもつことが可能なシステム環境)で同時に動作するプロセスやスレッドが特定されないシステムにおいては、CPU等に対する必要なクロック周波数をどのように決めるかが問題となる。

## 【0135】

シングルスレッドのシステムでは、予めそのスレッドで必要となるCPUやDSP、バス等のクロックが判っているので、それらの動作を満足するクロック周波数に設定しておけば良いが、マルチスレッドのシステムでは、予め各クロック周波数を特定することが出来ない。そこで、例えば、CPU等の稼動状況を計測してクロック周波数を動的に可変制御する方法が採られるが、その応答時間の遅れが問題となる場合には、例えば、必要な性能が不足したことが計測により判明してからクロック周波数が上がることになる。そのため、負荷の重いスレッドやプロセスが起動された場合に、一時的に性能が足りなくなる虞があったり、また、スレッドやプロセスが終了した場合に、CPU等の性能に余力があることが計測後に判明してからクロック周波数が下げられるので、その間無駄な電力を消費してしまうことになる。

## 【0136】

そこで、上記形態(II)を採用する場合において、以下では、下記に示す制御を採用する。

## 【0137】

- ・プロセス又はスレッドで必要とされるクロック周波数を予め登録しておくこと
- ・システムにおいて動作中の各プログラムに関して、現在起動されている全てのプロセス又はスレッドに係る周波数の加算総計値を周波数マネージャが算出して、クロック周波数の設定管理を行うこと
- ・起動されたプロセスやスレッドが終了した場合には、それらに係る周波数を周波数マネージャが周波数の加算総計値から減算することでクロック周波数を更新すること。

## 【0138】

つまり、各プロセスやスレッドで必要となる周波数を、事前にデータベースや参照テーブル、ヘッダ等に登録しておいて、現時点で起動されている全てのプロセス又はスレッドで必要な周波数を満足する周波数を算出するために、各プロセス又はスレッドが起動される前に必要な周波数を足し込み、また、プロセス又はスレッドが終了した直後には、その分の周波数を総和値から減算することによって常に最適な周波数を設定して、該周波数のクロックをCPU、DSPやメモリ、バス等に供給する。尚、適用されるシステムについては、例えば、下記の形態が挙げられる。

## 【0139】

- ・動的(ダイナミック)な可変設定が可能なクロックが、CPU等、システム上の機能においてそれぞれ独立している構成形態
- ・一部の機能のみについてクロック周波数の可変設定が可能な構成形態
- ・全ての機能や複数の機能をまとめてクロック周波数の可変設定を行うことができる構成形態。

## 【0140】

以下では、簡単化のため、一つのクロックの周波数を動的に可変できる構成形態を例にして、図11を用いて説明する。

## 【0141】

OSで必要となるクロック周波数又はシステムの機能を保証するために必要な最小限のクロック周波数を「A0」とする。尚、ここで、「A0」としては、OSに必要な周波数値と、システム保証に必要な周波数値のうち、大きい方をとる。

## 【0142】

スレッド1(本例では、スレッドについて説明するが、以下の説明で「スレッド」と表記した部分を、プロセスと適宜に置換しても良い。)で必要となるクロック周波数は、予め計測された「A1」であり、同様に、スレッド2で必要となるクロック周波数が「A2」、スレッド3について「A3」、以下「A4」、「A5」、…という具合に、全てのスレッドに関して必要とされる各クロック周波数が予め計測されて判明しているものとする。

## 【0143】

OSのみが動作している場合のクロック周波数は「A0」であり、ここで、スレッド1を起動しようとする場合にスレッド1で必要となるクロック周波数A1をA0に足した「 $A0 + A1$ 」が設定され、その周波数のクロックがCPU等に供給されてからスレッド1を起動する。さらに、スレッド2を起動する場合は現状の「 $A0 + A1$ 」に対して、スレッド2で必要となるクロック周波数A2を加算した「 $A0 + A1 + A2$ 」にクロック周波数を上げてからスレッド2を起動する。このようにスレッドを起動する場合にはそのスレッドで必要となる周波数を加算した周波数となるようにクロックを制御してからスレッドを起動させる。予め必要なクロック周波数に上昇させてからスレッドを起動するので、新たに起動されたスレッドによって性能が不足することはない。尚、実際に設定される周波数が連続量として任意に変更し得るとは限らないので、例えば、ハードウェア構成によっては、「 $A0 + A1 + A2$ 」のクロック周波数をそのまま正確に設定できない場合もあるが、そのような場合には「 $A0 + A1 + A2$ 」を下回らない周波数に設定すれば良い（「 $A0 + A1 + A2$ 」を越える周波数の選択肢のうち、それに最も近い値を選ぶか、あるいは、周波数の異なる複数のクロックを時間的に組み合わせて近似値を使用する等。）。

#### 【0144】

逆に、スレッド1乃至3が起動されていて（そのときのクロック周波数は「 $A0 + A1 + A2 + A3$ 」である。）、スレッド2を終了させる場合にはスレッド2を終了させた後、該スレッド2で必要であったクロック周波数A2を減算した「 $A0 + A1 + A3$ 」にクロック周波数を落とす。さらにスレッドA1を終了させた場合には「 $A0 + A3$ 」にクロック周波数を落とす。このようにスレッドを終了させてからクロック周波数を下げることにより、性能不足等が発生しないように防止し、また、スレッドの終了直後にその分のクロック周波数を落とすことで、直ちに消費電力を下げる事が可能となる。

#### 【0145】

尚、上記した方法は、可変できるクロック周波数のソース（クロック信号発生手段）が複数ある場合にも同様に適用することができ、別のクロックに対する必要周波数（B0、B1、B2、・・・やC0、C1、C2、・・・等）を、スレッド又はプロセスの起動又は終了に応じて各別に加算し又は減算すれば良い。

#### 【0146】

上記に説明した構成によれば、例えば、下記に示す利点が得られる。

#### 【0147】

- ・ハードウェアだけで把握できない情報を用いて、周波数や電源電圧の制御を行うことにより、性能を向上させたり、消費電力を低下させることができる
- ・アプリケーション等によって、最適な周波数や周波数の範囲を指定することが可能となり、不用意な周波数低下による性能劣化や、不必要な周波数の増大による消費電力の増加を避けることができる。

#### 【0148】

- ・複数のアプリケーションが動作しており、バックグラウンドで他のジョブが処理されているような状況においても、周波数マネージャがその状況を適正に把握して管理することで、最適な周波数設定を行うことが可能である。

#### 【0149】

- ・スレッド又はプロセスが起動される前に、必要となるクロック周波数に上げることができるので起動直後に性能不足が起きない。また、スレッド又はプロセスを終了する場合は、その終了直後にクロック周波数を下げることができるので、すぐに消費電力を低減することができる（各スレッドやプロセスで必要となるクロック周波数を予め測定してあるので、必要なクロック周波数を過不足なく供給することができ、厳密な電力制御が可能となる。）。

#### 【0150】

尚、本発明の適用において、前記した装置やこれに用いるプログラムに限らず、上記した動作周波数の制限機能を有するプログラムを記録した情報記録媒体（情報処理機器に装着してそのプログラムをメモリロードできるようにした各種の記録媒体や記憶媒体）や、

制御対象に係る動作周波数の制限法等に幅広く適用することができる。

【図面の簡単な説明】

【0151】

【図1】本発明に係る基本構成を示す概念図である。

【図2】周波数制限について処理例を示すフローチャート図である。

【図3】図4乃至図9とともに、本発明に係る実施例を示すものであり、本図は要部の構成例を示すブロック図である。

【図4】タイマ12の構成例を示す図である。

【図5】カウンタ13の構成例を示す図である。

【図6】カウンタ13の動作を説明するためのタイミングチャート図である。

【図7】クロック供給回路15の構成例を示す図である。

【図8】図9とともに制御回路14について説明するための図であり、本図は動作説明用のフローチャート図である。

【図9】回路構成の一例を示す図である。

【図10】周波数マネージャによる制御例の説明図である。

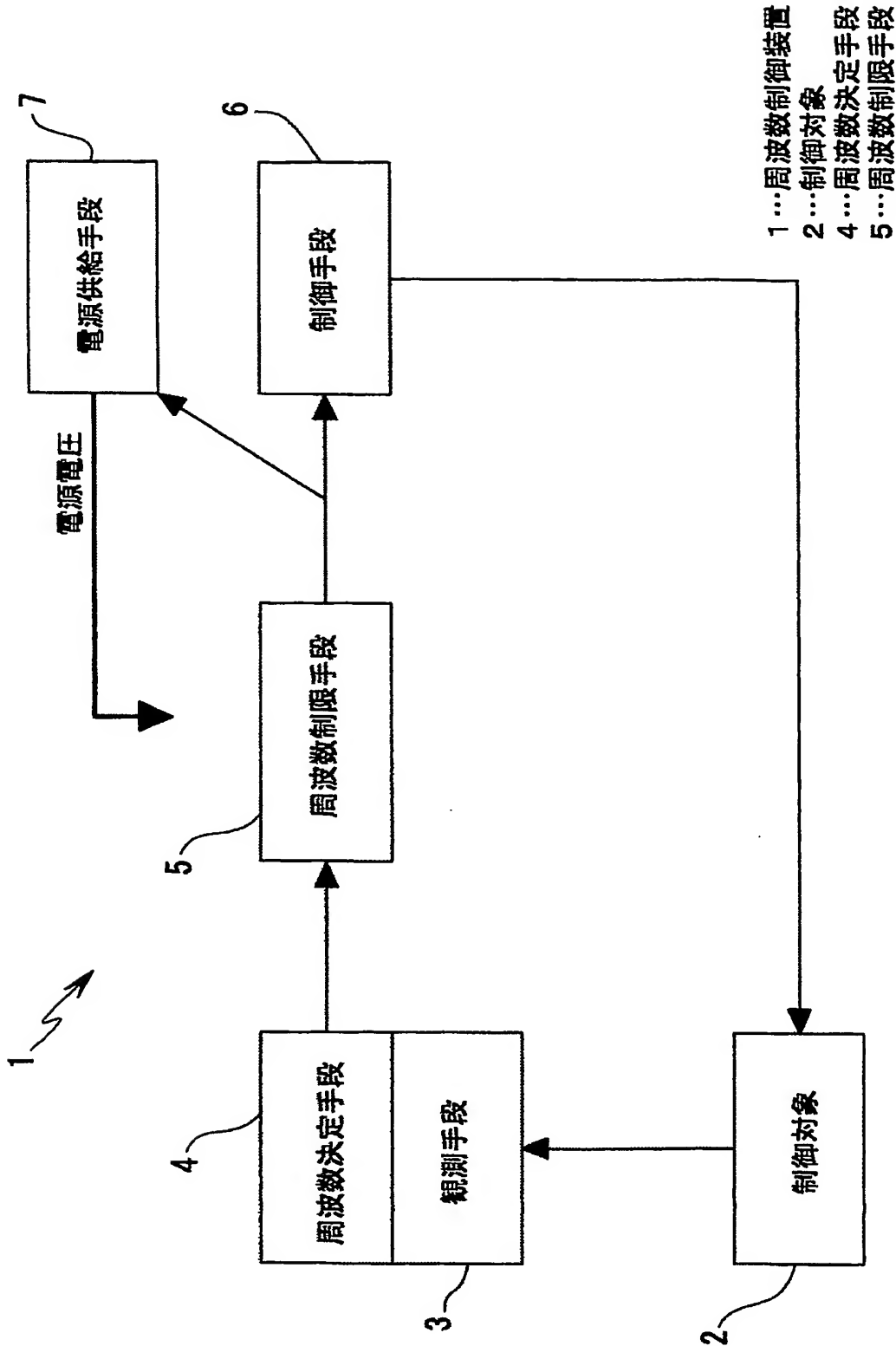
【図11】スレッドの生成、終了に伴う周波数設定例について説明するための概念図である。

【符号の説明】

【0152】

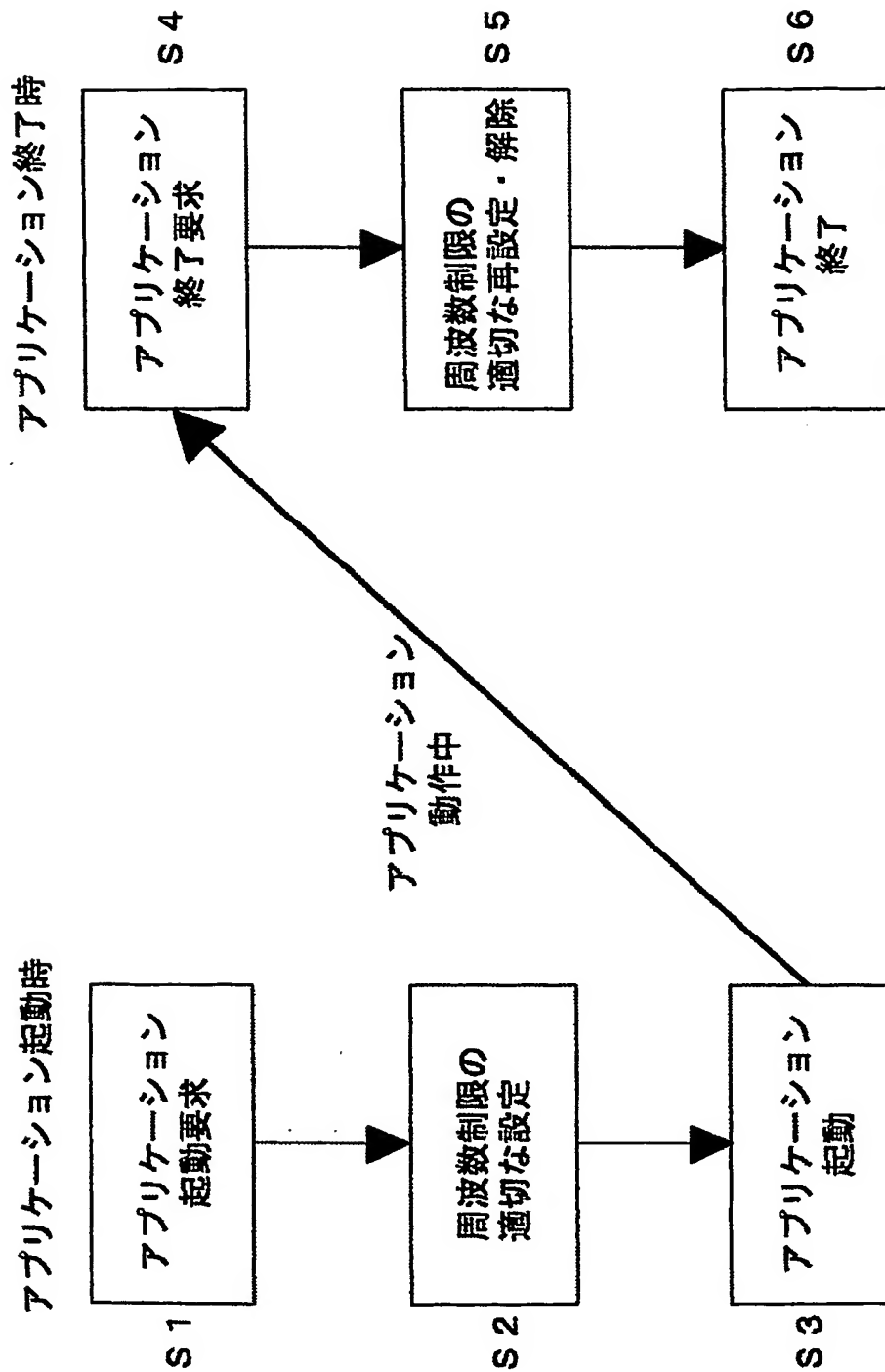
1…周波数制御装置、2…制御対象、4…周波数決定手段、5…周波数制限手段、9…周波数管理手段、10…情報処理装置、17…閾値設定手段、17U、17L…周波数設定用レジスタ、18…周波数指定手段

【書類名】 図面  
【図 1】

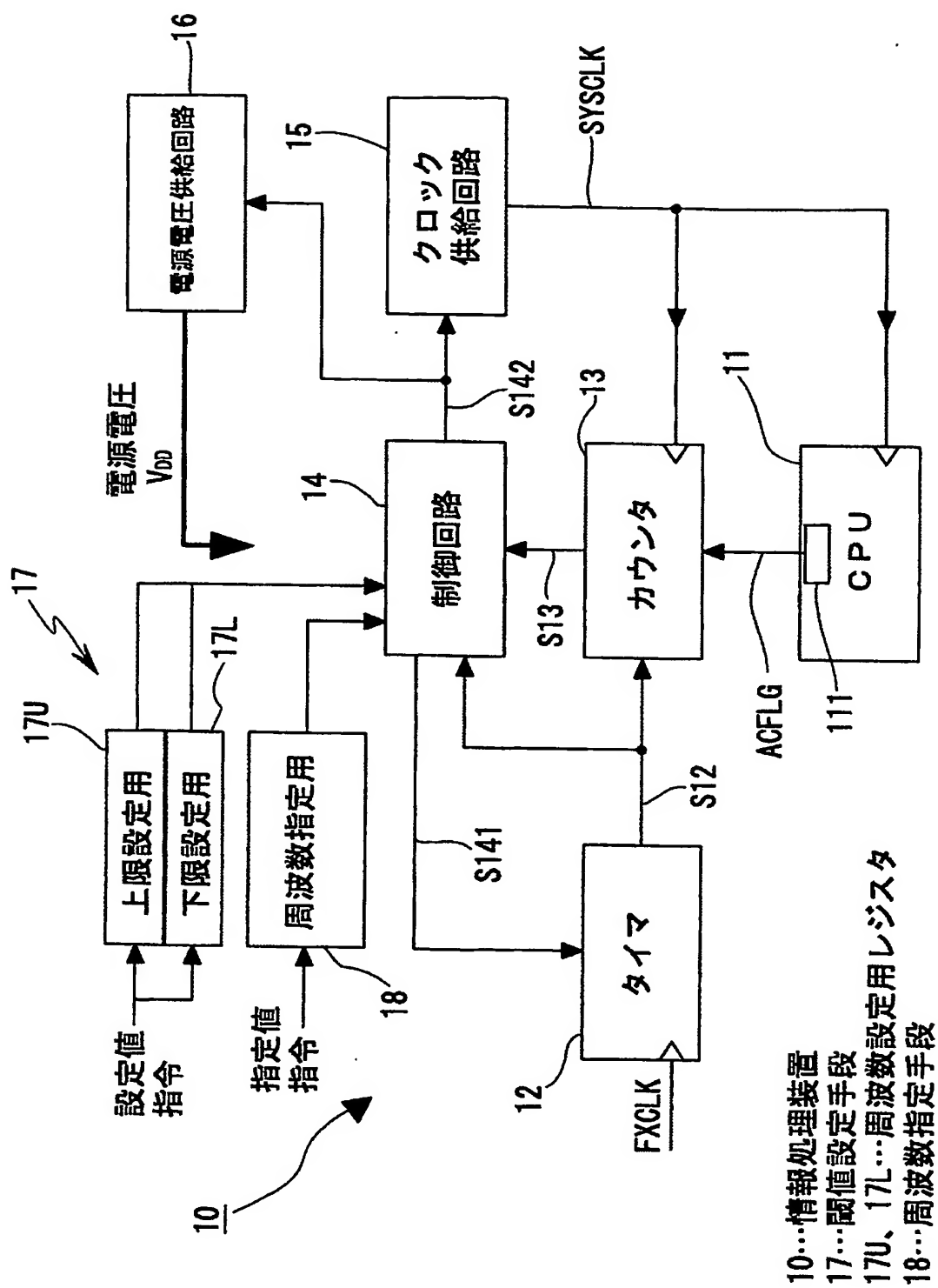




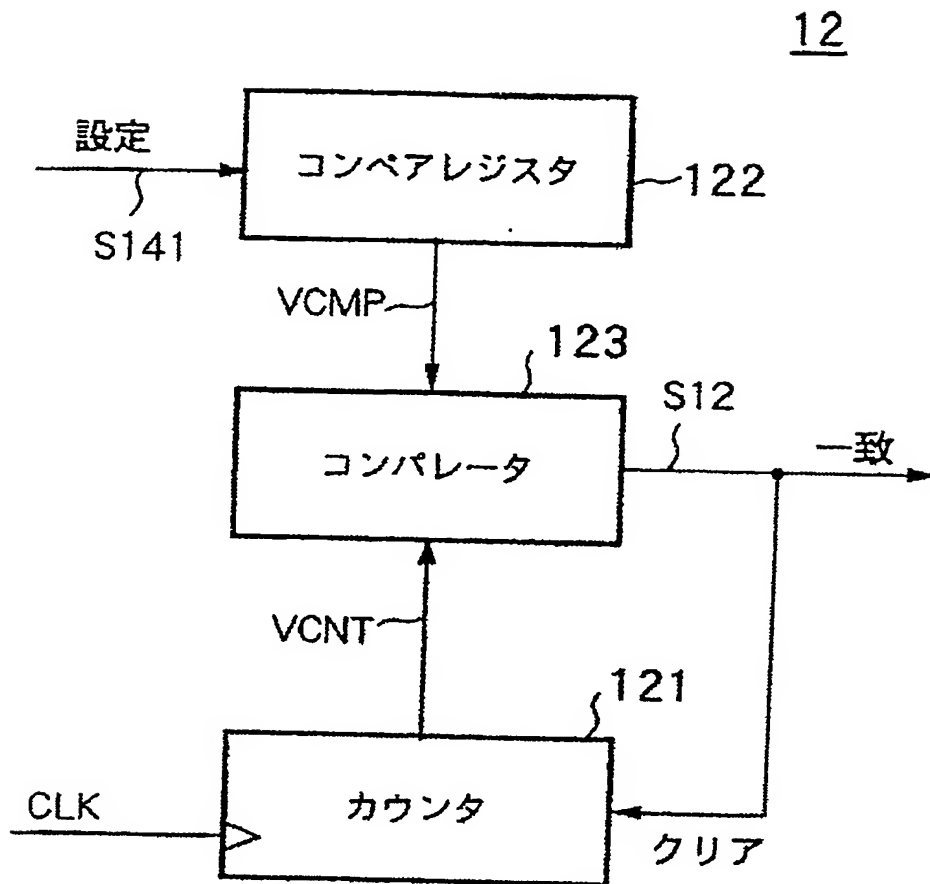
【図 2】



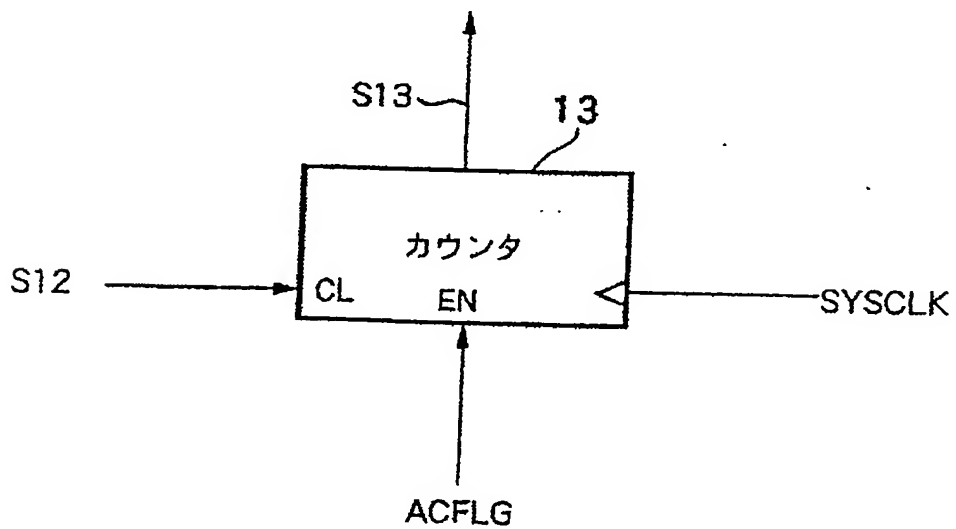
【圖 3】



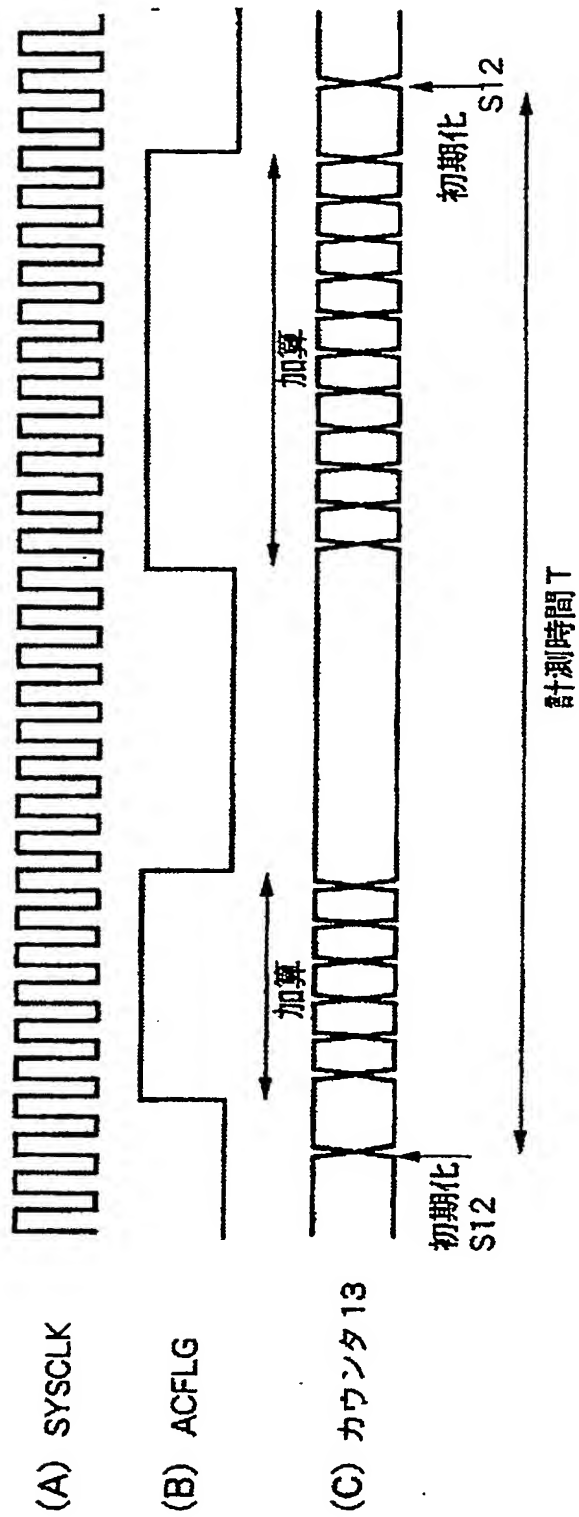
【図 4】



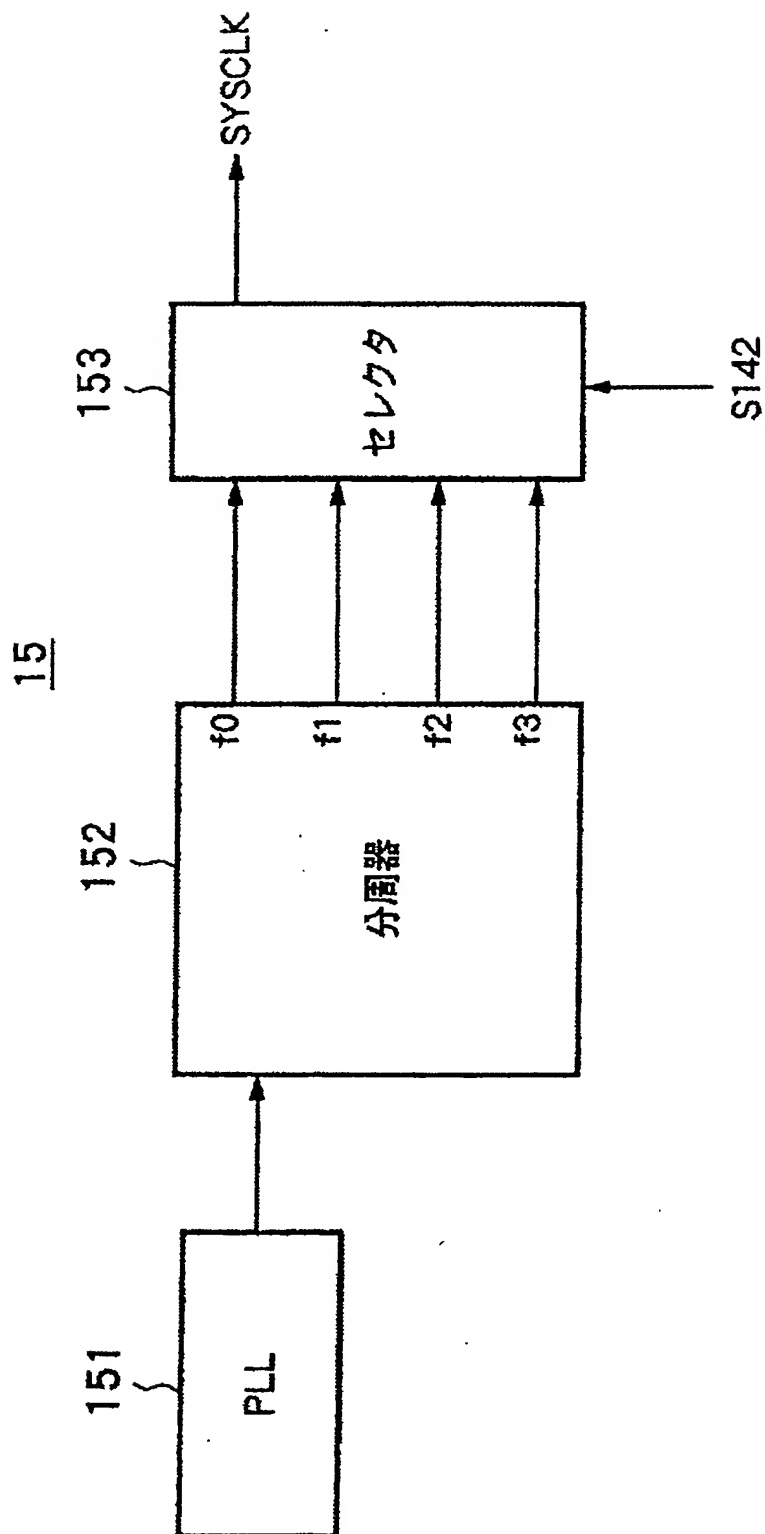
【図 5】



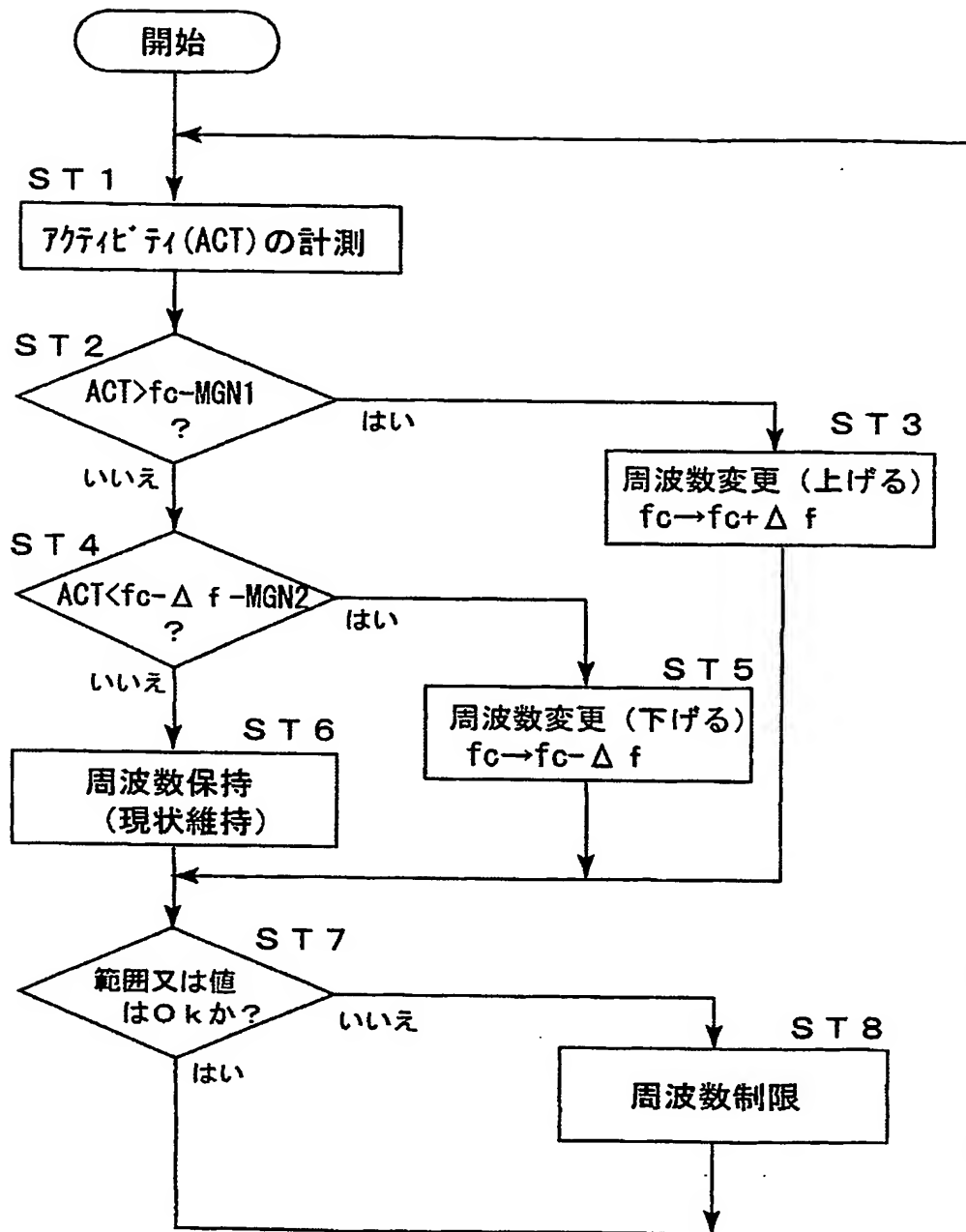
【図 6】



【図 7】

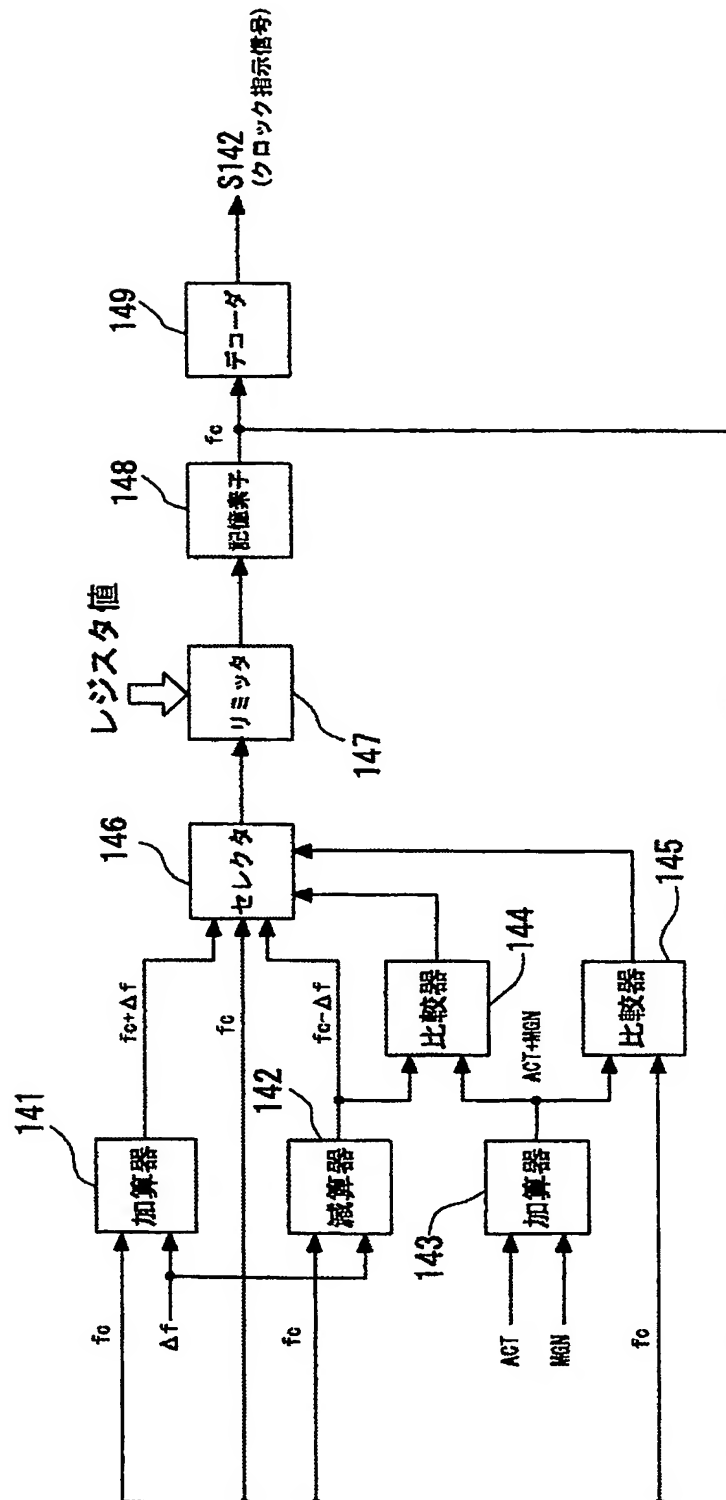


【図 8】

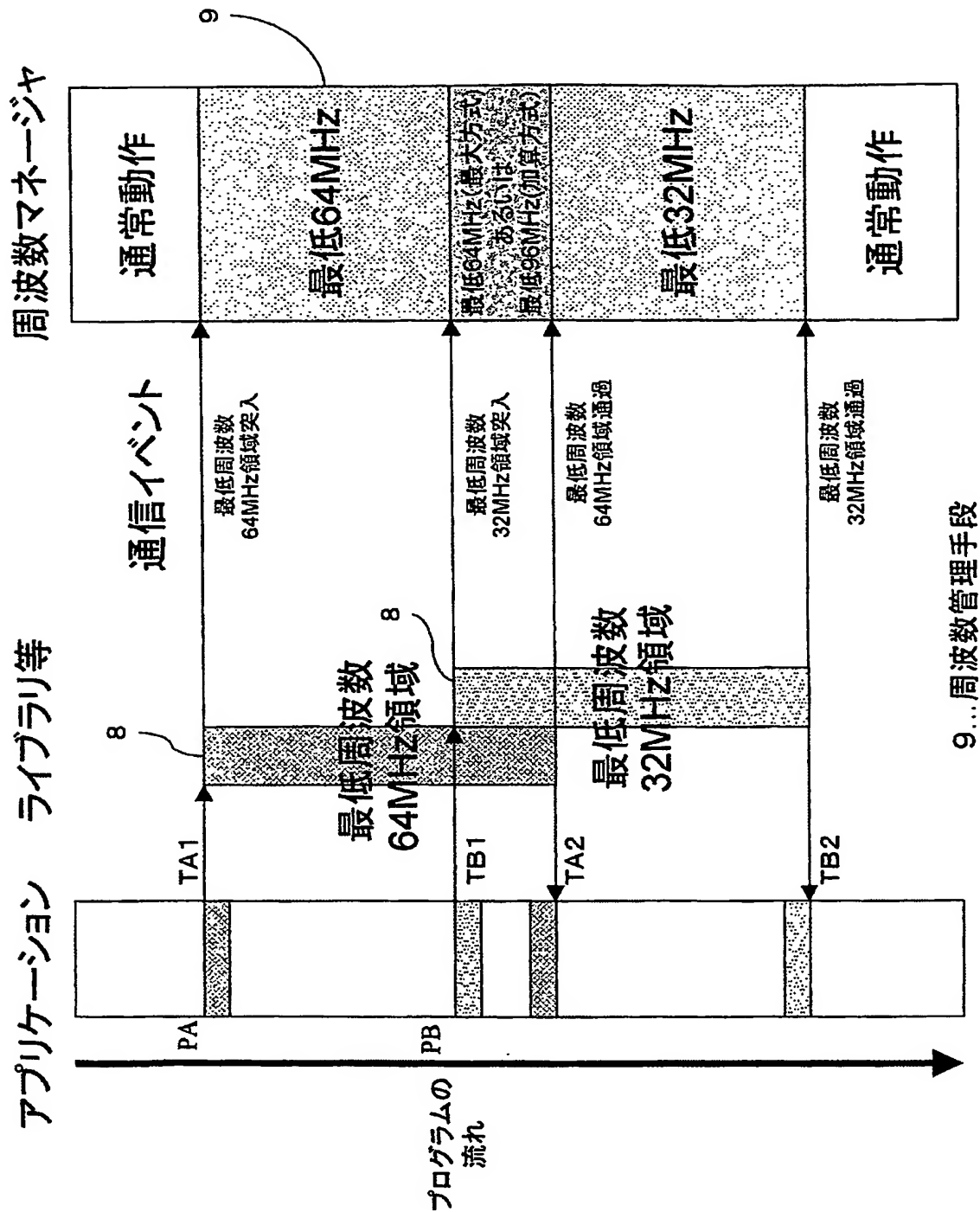


【図 9】

14

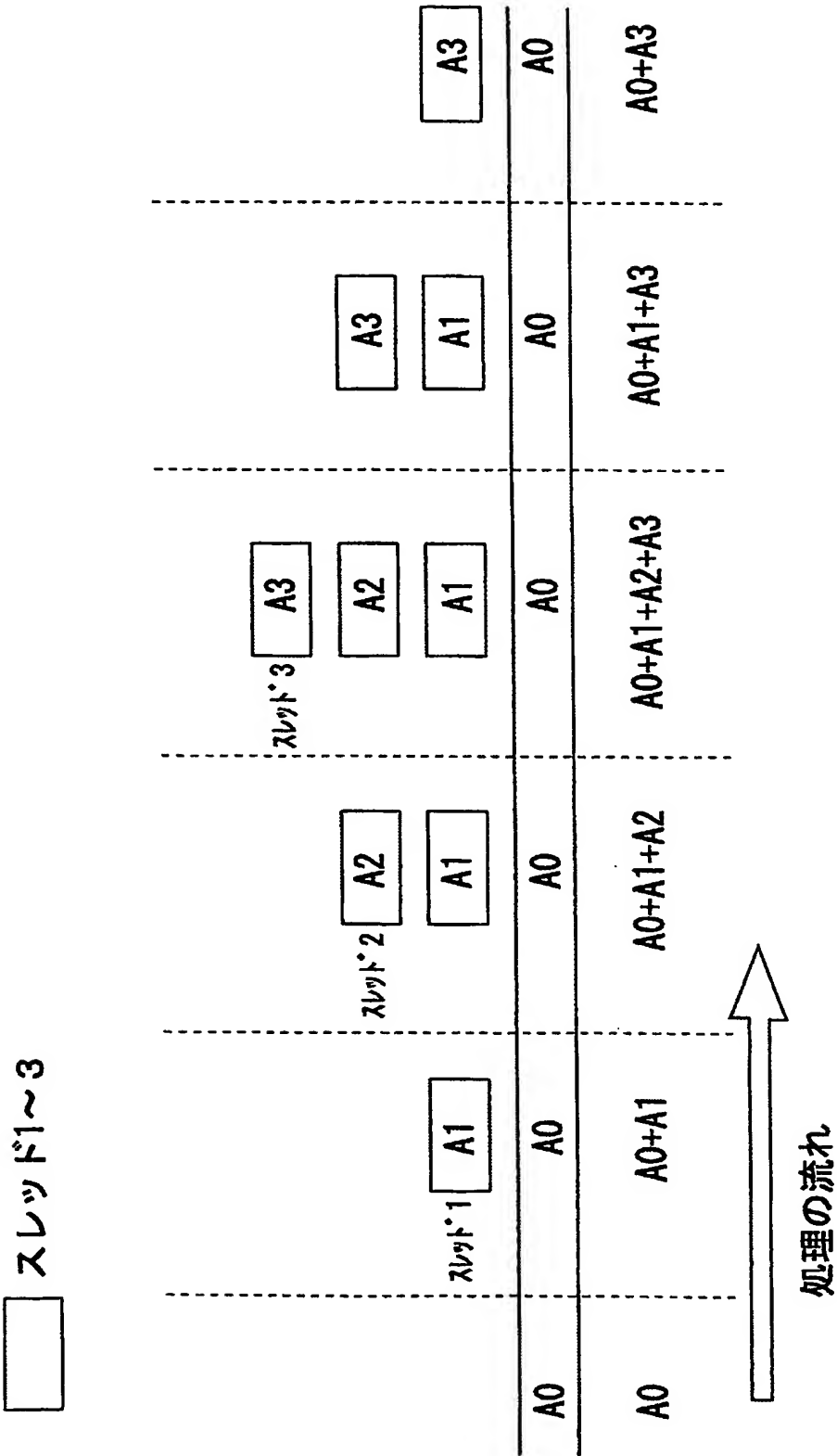


【図10】





【図 11】



## 【書類名】 要約書

## 【要約】

【課題】 制御対象の稼動状態の観測結果に応じた動作周波数制御において、動作周波数を制限したり許容値等を指定できるようにし、性能の向上や省電力化等を実現する。

【解決手段】 周波数制御装置 1 において、可変制御される周波数に基づいて動作する制御対象 2 の稼動状態を観測する観測手段 3 と、稼動状態に応じてクロック周波数を決定する周波数決定手段 4 と、周波数決定手段 4 により決定されるクロック周波数について、その範囲又は値を制限する周波数制限手段 5 を設ける。制御対象 2 の稼動状態に応じて決定される周波数が許容範囲である場合には、該周波数に従って制御を行うが、周波数が許容範囲外である場合等において周波数を制限できるように構成した。

【選択図】 図 1

## 認定・付加情報

特許出願の番号	特願 2003-272333
受付番号	50301139820
書類名	特許願
担当官	第七担当上席 0096
作成日	平成 15 年 7 月 14 日

## &lt;認定情報・付加情報&gt;

## 【特許出願人】

【識別番号】	000002185
【住所又は居所】	東京都品川区北品川 6 丁目 7 番 35 号
【氏名又は名称】	ソニー株式会社

## 【代理人】

申請人

【識別番号】	100069051
【住所又は居所】	東京都中央区八丁堀 3 丁目 17 番 12 号
【氏名又は名称】	小松 祐治

## 【選任した代理人】

【識別番号】	100116942
【住所又は居所】	東京都中央区八丁堀 3-17-12 小松特許事務所
【氏名又は名称】	岩田 雅信

特願 2 0 0 3 - 2 7 2 3 3 3

出 願 人 履 歴 情 報

識別番号

[ 0 0 0 0 0 2 1 8 5 ]

1. 変更年月日

1 9 9 0 年 8 月 3 0 日

[変更理由]

新規登録

住 所

東京都品川区北品川 6 丁目 7 番 3 5 号

氏 名

ソニー株式会社